Int J Appl Earth Obs Geoinformation 88 (2020) 102086

Contents lists available at ScienceDirect

Int J Appl Earth Obs Geoinformation

journal homepage: www.elsevier.com/locate/jag

A multi-level context-guided classification method with object-based convolutional neural network for land cover classification using very high resolution remote sensing images

Chenxiao Zhang[a], Peng Yue[b,c,d,*], Deodato Tapete[e], Boyi Shangguan[b], Mi Wang[a], Zhaoyan Wu[b]

[a] Wuhan University, State Key Laboratory of Information Engineering in Surveying, Mapping and Remote Sensing (LIESMARS), 129 Luoyu Road, Wuhan, Hubei, 430079, China
[b] Wuhan University, School of Remote Sensing and Information Engineering, 129 Luoyu Road, Wuhan, Hubei, 430079, China
[c] Wuhan University, Hubei Province Engineering Center for Intelligent Geoprocessing (HPECIG), 129 Luoyu Road, Wuhan, Hubei, 430079, China
[d] Collaborative Innovation Center of Geospatial Technology, 129 Luoyu Road, Wuhan, Hubei, 430079, China
[e] Italian Space Agency (ASI), Via del Politecnico snc, 00133, Rome, Italy

ARTICLE INFO

Keywords:
VHR image
Object-based image classification
Remote sensing classification
Convolutional neural network
Deep learning

ABSTRACT

Classification of very high resolution imagery (VHRI) is challenging due to the difficulty in mining complex spatial and spectral patterns from rich image details. Various object-based Convolutional Neural Networks (OCNN) for VHRI classification have been proposed to overcome the drawbacks of the redundant pixel-wise CNNs, owing to their low computational cost and fine contour-preserving. However, classification performance of OCNN is still limited by geometric distortions, insufficient feature representation, and lack of contextual guidance. In this paper, an innovative multi-level context-guided classification method with the OCNN (MLCG-OCNN) is proposed. A feature-fusing OCNN, including the object contour-preserving mask strategy with the supplement of object deformation coefficient, is developed for accurate object discrimination by learning simultaneously high-level features from independent spectral patterns, geometric characteristics, and object-level contextual information. Then pixel-level contextual guidance is used to further improve the per-object classification results. The MLCG-OCNN method is intentionally tested on two validated small image datasets with limited training samples, to assess the performance in applications of land cover classification where a trade-off between time-consumption of sample training and overall accuracy needs to be found, as it is very common in the practice. Compared with traditional benchmark methods including the patch-based per-pixel CNN (PBPP), the patch-based per-object CNN (PBPO), the pixel-wise CNN with object segmentation refinement (PO), semantic segmentation U-Net (U-NET), and DeepLabV3+(DLV3+), MLCG-OCNN method achieves remarkable classification performance (> 80 %). Compared with the state-of-the-art architecture DeepLabV3+, the MLCG-OCNN method demonstrates high computational efficiency for VHRI classification (4–5 times faster).

1. Introduction

Very high resolution images (VHRI; < 1 m) are increasingly available from optical sensors (e.g. WorldView-3, GeoEye-1, QuickBird and Gaofen-2). The fine contextual information and complex spatial characteristics that these images convey offer rich spatial details for advanced land cover analysis (Chen et al., 2019a; Vetrivel et al., 2018). On the other side, in addition to known challenges for classification (e.g. intra-class spatial or spectral heterogeneity, and vice versa inter-class similar spatial or spectral patterns; Zhao and Du, 2016), VHRI introduce new ones for image interpretation. For example, identification of small rectangular parcels of glass as a distinct class is difficult because pixels and objects belonging to glass can be assigned to building class if they are located on building roofs (e.g. skylights) or, alternatively, to car class if the pixels belong to car window. Consequently, the high intra-class variability and low inter-class disparity, plus the semantic diversity, make VHRI classification challenging.

From pixel statistical analysis methods (Ichoku and Karnieli, 1996), scholars moved to object-based image analysis (OBIA) to segment images into “meaningful” objects (superpixels) with relatively discrete spatial pattern, high interior homogeneity and discreteness. Most of object-based image classification (OBIC) approaches follow a

* Corresponding author.
E-mail address: pyue@whu.edu.cn (P. Yue).

https://doi.org/10.1016/j.jag.2020.102086
Received 22 October 2019; Received in revised form 27 January 2020; Accepted 10 February 2020

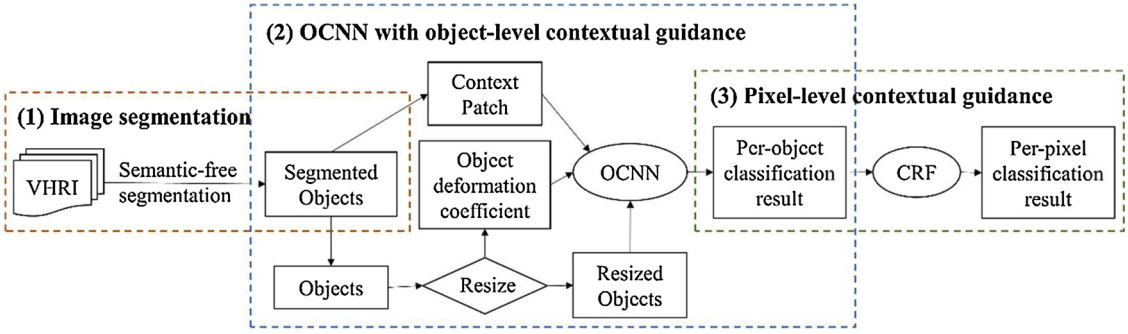

Fig. 1. Workflow of MLCG-OCNN method: (1) image segmentation, (2) object-based convolutional neural network (OCNN) with object-level contextual guidance, (3) pixel-level contextual guidance by means of Conditional Random Field (CRF).

"segmentation-classification" mode. First segmentations, such as Multi-Resolution, Mean-Shift, and Quadtree-Based approaches, are performed. Then, during the classification stage, hand-crafted features of segmented objects are fed into supervised or unsupervised classifiers. Although OBIC overcomes the salt-and-pepper effect that is common in per-pixel approaches (Chen et al., 2019a), the manually-designed rule set of hand-crafted features fails to achieve a satisfactory performance on VHRI, because the hand-crafted features are limited in representing rich textural features and understanding complex geometric details.

Deep learning proves very promising for land use and land cover classification (Luus et al., 2015; Weng et al., 2017; Zhang et al., 2018a), scene classification (Castelluccio et al., 2015), change (Zhang et al., 2016) and object detection (Cheng et al., 2016; Zhang et al., 2018c). Multi-layer artificial Convolutional Neural Network (CNN) allows automatic extraction of high-level features from labeled images. By means of convolutional kernels at multi-levels operating over upper-level feature maps, high-level features are extracted hierarchically through the network. The back-propagation strategy helps CNN adjust its network parameters automatically. The high generalization capacity of CNN outstands other machine learning algorithms and makes CNN the most mature and widely used deep learning framework (LeCun et al., 2015).

CNN-based land cover classification often follows a patch-based strategy (Lv et al., 2018; Zhang et al., 2018a). The patch-based strategy applies a moving window with a fixed size on each pixel to generate overlapping patches (Nguyen et al., 2013; Marmanis et al., 2016; Sharma et al., 2017). Then patches are fed into a CNN, which is composed of two functional parts. The first part of the CNN consists of multiple stacked convolutional and pooling layers that are used for feature extraction. The second part is usually implemented by a stack of fully connected layers with the SoftMax layer at the end to generate a probability distribution over different classes. Alternative algorithms have been used at the second part such as SVM (Agarap, 2017), XGBoost (Ren et al., 2017), and forest random (Richmond et al., 2015). Owing to its deep feature extraction ability, patch-based CNN exhibits superior performance in extracting high-level features than methods based on hand-crafted features (Othman et al., 2016). In addition, a variety of multi-scale CNNs have been presented to overcome the limited perception field of single-scale patches. They were proven to be more effective than the single-scale CNN (Hu et al., 2015; Zhao et al., 2015; Alhichri et al., 2018; Wu et al., 2019a, 2019b). However, fine tune of patch size is tricky in "object segmentation - patch generation - patch prediction - object labeling" processes (e.g. Lv et al., 2018). If the patch size is too large, more than one object can be observed in a single patch, which leads to noise disturbance. Conversely, if the size is too small, the distinguishable characteristics of objects may not be captured. Consequently, a better choice is to feed objects straightforwardly into CNNs to overcome the limited representation ability of patches (Zhang et al., 2018d). However, objects must to be reshaped in different shapes and scales into the same size, because CNN requires a fixed size of input images. This operation causes the loss of object shape and scale information. Furthermore, relationships of the surrounding neighbors (i.e. contextual information) should also be considered for accurate object classification. Therefore, further investigations are needed to minimize the loss of shape and scale information during the data preprocessing stage and the incorporation of contextual information.

In this paper, a multi-level context-guided classification method with object-based convolutional neural network (MLCG-OCNN) is proposed. Objects with fine boundary and high internal compactness are firstly segmented as functional units by means of semantic-free segmentation. Instead of extracting features from patches to represent objects, MLCG-OCNN utilizes objects and context patches as inputs. A deep CNN fusing high-level features from independent spectral patterns, geometric characteristics, and object-level contextual information is developed for per-object classification. An object-as-analysis unit perspective is adopted, wherein object-level features can be best retained. The object geometric characteristics are kept, by combining the object mask strategy with an object deformation coefficient that measures the distortion of objects and proves effective to achieve contour-preserving results with high accuracy. Finally, a conditional random field (CRF) graph model is employed to explore the contextual information of neighboring pixels to further improve the classification results. The effectiveness and computational efficiency of the MLCG-OCNN method are compared with 5 benchmark methods: Patch-based per-pixel CNN (PBPP), Patch-based per-object CNN (PBPO), Object-based CNN (PO), semantic segmentation U-Net (U-NET), and DeepLabV3+ (DLV3+).

2. Methodology

Fig. 1 shows the three-step process of MLCG-OCNN method. Step 1: VHRI is segmented into objects by a semantic-free segmentation algorithm. Step 2: the per-object classification is run by means of context-guided object-based CNN (OCNN). Specifically, contour-preserving objects are clipped from images according to a mask policy. Object-oriented context patches are then produced by means of masking windows with flexible sizes on each object. At the same time, the object deformation coefficient is derived during the object resize operation, and then it is used as a supplement of geometric characteristics for object discrimination. Finally, the deep independent object features and contextual information (i.e. the features extracted from context patches) are fused in the proposed OCNN. Step 3: the per-object classification output is further processed by means of Conditional Random Field (CRF) for per-pixel refinement with pixel-level contextual guidance.

2.1. Image segmentation

For patch-based object classification methods, since CNN requires a fixed size for input images, patches holding different proportions of

backgrounds have negative impacts on object classification results (Chen et al., 2019a). Algorithms (e.g. SEEDS: Superpixels extracted via energy-driven sampling; Van et al., 2015), that segment images into objects with similar sizes and shapes, were adopted in existing object-based classification studies (Lv et al., 2019). However, these algorithms produce too many fragmented objects. Large objects are broken into pieces with similar sizes, ignoring their geometric boundaries or integrity as functional units. Continuously distributed areas such as long strips of roads and large areas of building roofs are cut into fragmented pieces, which result in the loss of object-level geometric information. MLCG-OCNN method discards the patch-based strategy and takes the object integrity as a priority. In this paper, we use Multi-resolution segmentation (MRS; Baatz and Schäpe, 2000), given that it produces results with meaningful objects (Neubert et al., 2008). The effect of different segmentation scales on classification results has been intentionally tested.

2.2. OCNN with object-level contextual guidance

After the image segmentation, a feature-fusing Object-based CNN (OCNN) is developed for per-object classification. The proposed OCNN receives three inputs, i.e. two images representing the labeled object and the context patch, respectively, and 1-D vector representing the object deformation coefficient (Fig. 2).

2.2.1. Object cropping

The minimum bounding rectangle (MBR) mask policy has been used widely to crop objects from images. A cropped image patch usually contains both the object and the background. A mix of different proportions of object and background have a negative effect on object feature extraction. A large proportion of an object in a patch contains more object pixels, while a small proportion of an object in a patch contains more background pixels that result in an excessive attention on the surrounding environments. Moreover, cropped patches generated by the mask policy are hard to provide geometric characteristics (i.e., shape and scale) of objects for object discrimination. Therefore, we propose an object boundary mask policy to crop fine contour-preserving objects from images. Patches are firstly masked by object MBR. Then, pixels falling inside the object boundary keep their original spectral values, while pixels falling outside the object boundary are assigned to zero value.

2.2.2. Object deformation coefficient

The object-level geometric characteristics are crucial for object discrimination, especially for inter-class objects having similar interior texture patterns while showing different shapes and sizes. For example, it is difficult to discriminate asphalt-covered roofs from asphalt-covered roads in terms of the spectral textures. The problem can be solved if geometric characteristics are taken into consideration. Although both roads and roofs are covered by grey-smoothed asphalt surfaces, roads can be easily discriminated based on their strip shape, while objects with approximately square shape are more likely to be roofs.

The loss of object geometric information during the resizing operation is not yet addressed in the literature. Therefore, MLCG-OCNN includes the object deformation coefficient. Assuming that MBR defining the original shape of an object O has a length of O_{BB_l} and a width of O_{BB_w}, CNN accepts input images with the specific size of $(Input\_Obj_l, Input\_Obj_w)$. Usually $Input\_Obj_l$ is equal to $Input\_Obj_w$. The object O is resized from (O_{BB_l}, O_{BB_w}) to $(Input\_Obj_l, Input\_Obj_w)$ by means of the resizing function. The deformation coefficient (C_{O_x}, C_{O_y}) for the object O is defined as follows:

$$(C_{O\_x},\ C_{O\_y}) = (O_{BB\_l}/Input\_Obj_l,\ O_{BB\_w}/Input\_Obj_w) \quad (1)$$

2.2.3. Object-level contextual guidance

Although the combination of deep features extracted from the fine contour-preserving object and the object deformation coefficient is effective for discriminating independent objects, it mainly focuses on features within object boundaries while it ignores the neighboring information of the objects. For instance, both car windshield windows and building roof windows have strong light reflectivity and high transparency, and they also have similar interior spectral textures and geometric characteristics. Training a CNN fed with these cropped images will lead to a poor classification performance.

In MLCG-OCNN, the object-oriented context patch is an image patch describing both the object and its surrounding environment. Fixed size windows are commonly used to crop context patches (Zhang et al., 2018b; Zhao et al., 2017; Fu et al., 2018; Lv et al., 2018). However, fixed window size has similar problem as the independent object cropping, i.e. the changing proportion of objects and their backgrounds. To overcome these shortcomings, we introduce the object-oriented context patch into the CNN for object-level contextual guidance. Windows with flexible sizes are used. Sizes of context patches are determined by corresponding objects in order to maintain the same reception field with objects. The size of each object context patch varies from object to object. Using a Minimum Bounding Box (MBB) for the object O, (O_{BB_x}, O_{BB_y}) is the central point of the MBB, and O_{BB_l} and O_{BB_w} denote the length and width of the MBB, respectively. Then, the

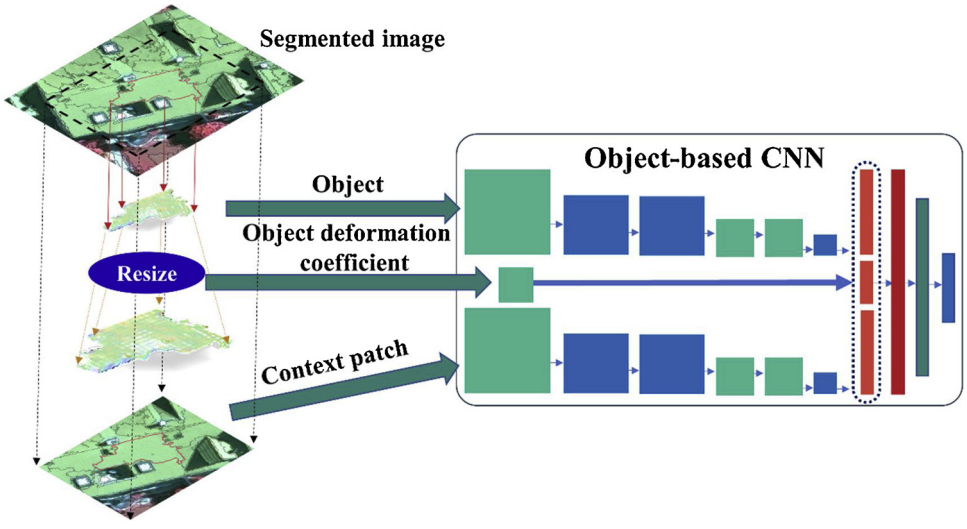

Fig. 2. Feature-fusing OCNN with object-level contextual guidance.

contextual patch $(O_{Cont\_x}, O_{Cont\_y}, O_{Cont\_l}, O_{Cont\_w})$ for the object O is defined as follows:

$$(O_{Cont\_x},\ O_{Cont\_y}) = (O_{BB_x},\ O_{BB_y}) \tag{2}$$

$$(O_{Cont\_l},\ O_{Cont\_w}) = (O_{BB_l} \times S_{re}, O_{BB_w} \times\ S_{re}) \tag{3}$$

where (O_{Cont_x}, O_{Cont_y}) is the central point and O_{Cont_l} and O_{Cont_w} are the length and width of the context patch. S_{re} controls the size of the reception field of the context patch. Small S_{re} produce small context patches containing poor neighboring information; high S_{re} produce context patches with large reception fields. However, too large S_{re} produce highly overlapping context patches, which hamper the classification performance and incur a huge computational cost. In Section 4.3.2 we discuss the effect of S_{re} for classification result.

The context patch is then resized to a square area $(Input\_Cont_l, Input\_Cont_w)$ defined as follows.

$$(Input\_Cont_l, Input\_Cont_w) = (Input\_Obj_l \times S_{re}, Input\_Obj_w, \times S_{re}) \tag{4}$$

where $Input\_Obj_l$ and $Input\_Obj_w$ represent the length and width of the resized object images. Therefore, the context patch for each object can be wrapped as an input to OCNN for object-level contextual guidance.

2.2.4. Object-based Convolutional Neural Network (OCNN)

A typical CNN is built upon stacked blocks composed by convolutional and pooling layers. Starting from the first convolutional kernel scanning across patch images, characteristics of the patch image are abstracted hierarchically as high-level features through a stack of convolutional-pooling blocks. A nonlinear activation function follows each convolutional layer to strengthen the non-linearity. Mathematically, the operation of the convolutional-pooling block is as follows:

$$f_i = Pool_i(F_{non\_linear}(f_{i-1} * W_i + b_i)) \tag{5}$$

where f_i and f_{i-1} represent feature maps for the i_{th} and $i-1_{th}$ convolutional-pooling block, respectively. The feature map of the layer f_{i-1} is convolved by W_i. W_i denotes weights of the current convolutional kernel. b_i represents biases following the convolution. The non-linearity function, i.e. F_{non_linear}, is operated over the convolution result to strengthen the non-linearity. Afterwards, the pooling layer ($Pool_i$) is applied using the max pooling method. After the feature extraction, a stack of fully connected layers is attached to the last pooling layer to learn non-linear combinations of extracted features. The fully connected layer starts with a fixed-length vector as the input, while the convolutional layer produces outputs in arbitrary sizes by taking input with arbitrary sizes. Consequently, typical CNNs demand input of images with a fixed size. To make the proposed OCNN work with datasets with variable inputs, a Spatial Pyramid Pooling (SPP) Layer is applied to bridge the non-fixed length of features and the fixed length requirement for input of the fully connected layers. Instead of using a sliding window over feature maps, SPP employs several spatial bins with sizes proportional to feature maps for feature pooling. In this way, the number of the extracted high-level features is fixed.

After the above steps, labeled objects, context patches, and the derived object deformation coefficients are fed into the feature-fusing OCNN. Built by parallel stacks of convolutional-pooling blocks, the OCNN allows for multi-input feature extraction and fusion, and learns object labeling. Parameters and structure of the model are further tuned empirically, as shown in the experimental section. After the training phase, the trained OCNN is used for per-object classification. The classification result is further refined in the following stage using the pixel-level contextual guidance.

2.3. Pixel-level context-guided classification

The object contour delineation using the traditional image segmentation method, i.e. MRS, ignores the rich contextual information among semantics-embedded pixels. For per-object classification results, contour delineation errors introduced by image segmentation can be refined at pixel-level. Therefore, the CRF is further introduced for per-pixel classification refinement.

Suppose that $X = \{x_1, x_2, \ldots, x_N\}$ is a list of random variables, each x_i corresponds to the pixel at location i in the image, and takes a value from a category set of $L = \{l_1, l_2, \ldots, l_k\}$. N is the total number of pixels in the image. k is the number of classes for the category set. $F = \{F_1, F_2, \ldots, F_N\}$ denotes the observed data sequence for the image, and F_i represents the spectral feature of pixel i. Then, the conditional random field (F, X) is formulated by means of a Gibbs distribution. The pair (F, X) is defined as follows:

$$P(XF) = \exp(-E(XF))/Z(F) \tag{6}$$

where $Z(F)$ is a normalizing factor that is computed as follows:

$$Z(F) = \sum_x exp(-E(X|F)) \tag{7}$$

$E(X)$ in Eq. (6) represents the Gibbs energy. The result is achieved by finding the minimum value of $E(X)$. The Gibbs energy of pairwise CRF takes the following form:

$$E(X) = \sum\nolimits_i \varnothing_i(x_i) + \sum\nolimits_{i,j} \varphi_{i,j}(x_i,\ x_j) \tag{8}$$

where the unary potentials $\varnothing_i(x_i) = -\log(P(X_i = x_i))$.

$P(X_i = x_i)$ represents the probability of the pixel i to take the label of x_i.

$\varphi_{i,j}(x_i,\ x_j)$ is the pairwise potentials defined as follows:

$$\varphi_{i,j}(x_i,\ x_j) = \mu(x_i,\ x_j)[w_1\exp(-\|p_i - p_j\|^2/2\sigma_\alpha^2 - \|I_i - I_j\|^2/2\sigma_\beta^2) + w_2\exp(-\|p_i - p_j\|^2/2\sigma_\gamma^2)] \tag{9}$$

where I_i and I_j are feature vectors for pixel i and j. p_i and p_j are pixel positions. $\|p_i - p_j\|^2$ and $\|I_i - I_j\|^2$ denote the spectral distance and spatial distance between pixel i and j, respectively. w_1, w_2, σ_α, σ_β and σ_γ are weight controlling parameters. The first term in Eq. (9) describes the degree of adjacent pixels in similar colors belonging to the same category. $\mu(x_i,\ x_j)$ is the label compatibility function. If $x_i \neq x_j$, then $\mu(x_i,\ x_j) = 1$; otherwise, $\mu(x_i,\ x_j) = 0$. This means that adjacent pixels in similar colors, yet assigned to different classes, should be penalized. In this way, the first term allows that nearby pixels with the similar color should have the same label. The second term depends only on the spatial distance between pixels, and it helps remove isolated regions.

The probability of pixels within an object boundary belonging to different classes is determined primarily according to the OCNN classification result. Then, the probability is used as the prior probability in the unary potentials. By performing the pixel-level context-guided CRF, the object boundary is finely modified, and the isolated small objects in the image are removed.

3. Experiments and analysis

3.1. Datasets

Two image pairs containing one training and one testing images from the following open datasets are used:

- Vaihingen Semantic Labelling dataset (ISPRS, 2013a) with spatial resolution of 0.09 m. Two sub-scenes (Vai-12 and 13) with a band combination of red, green and near-infrared were chosen. The ground reference maps include 5 categories: buildings, trees, low-vegetation, cars, and roads. The Vai-13 (Fig. 3a) and Vai-12 (Fig. 3b) images have 2817 × 2557 and 1921 × 2574 pixels and were used for training and testing, respectively.
- Potsdam Semantic Labelling dataset (ISPRS, 2013b) with spatial resolution of 0.05 m. Ground reference major categories of the

Fig. 3. ISPRS Vaihingen image pair (false color composite: red, green, near-infrared bands): (a) Vai-13 and (b) Vai-12 images used for training and testing the model, respectively.

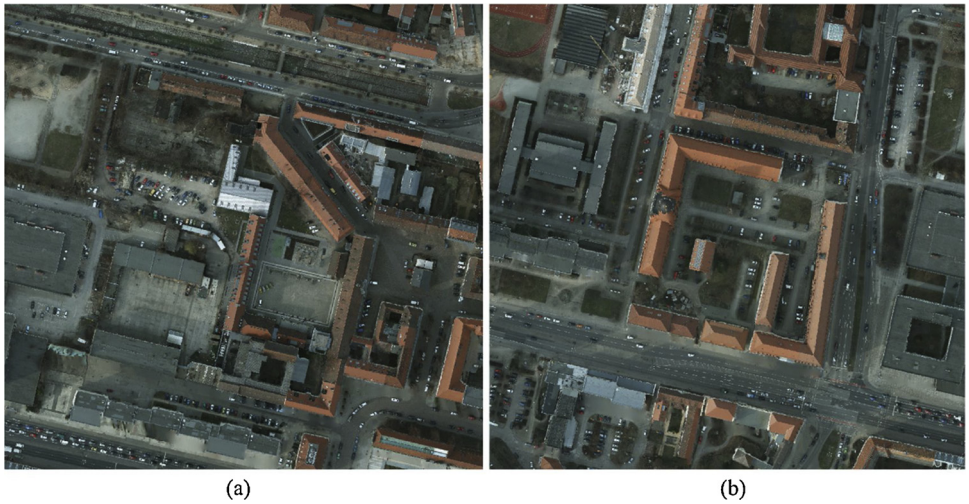

Fig. 4. ISPRS Potsdam image pair (RGB color composite: red, green and blue bands): (a) 7-12 and (b) 7-11 images used for training model and testing the model.

Potsdam dataset include roads, buildings, trees, low vegetation, cars, and clutters. Images 7–11 and 7–12 with three bands (red, green and blue), and associated reference maps, containing 6000 × 6000 pixels were used in the experiment for training (Fig. 4a) and testing the model (Fig. 4b), respectively.

In terms of data volume, the Potsdam images (each image contains 6000 × 6000 pixels) are larger than the Vaihingen images (the largest image contains 2817 × 2557 pixels). We intentionally exploit the difference in data volume (i.e. 6000 × 6000 pixels vs. 2817 × 2557 pixels) to assess the ability of MLCG-OCNN method and the selected benchmark methods (see Section 4.2.3) in dealing with datasets of different volumes.

We also intentionally selected subsets of images from two different datasets, because in practice it is often the case that there is only a limited number of training samples. While existing methods can achieve good performance using the full set of images and lots of training examples, it is still not clear how benchmark methods can perform in cases with a small set of images and a limited number of training samples. On one hand, classification performance of the proposed method on small datasets can be exploited. On the other hand, the assessment of performance gap among different benchmark methods is more significant if the experiment is carried out on small datasets, and thus the comparison is more straightforward.

3.2. Parameters and model structures

3.2.1. Segmentation parameters

The image segmentation was initially performed by means of MRS using eCognition 9 software. Among the three required parameters, i.e., shape, compactness, and segmentation scale parameter (SSP), SSP is the dominant parameter by controlling the average image object size. Specifically, a small SSP produces small objects with high internal compactness, while a large SSP results in large objects containing more pixels. For the object classification, a large number of small objects containing fewer pixels produced by a small SSP, will hamper the classification performance and require a high computational cost. Conversely, a relatively large SSP will result in oversized objects containing pixels belonging to different categories, which will negatively impact the deep feature extraction. To evaluate the effects of segmentation scale on the proposed OCNN, we employed different scales on both image pairs. SSPs with values of 20 and 50 were selected for the

Table 1
Numbers of objects for each class in the Vaihingen and Potsdam training images.

Class	Vaihingen		Potsdam	
SSP	20	50	40	100
Tree	12159	2543	1315	257
Low-vegetation	12853	2737	1799	366
Car	312	99	2461	768
Building	4184	1024	10355	2203
Road	4218	1027	13807	3020
Clutter	0	0	841	194
Total	33726	7430	30578	6808

Vaihingen images. Two larger SSPs (40 and 100) were employed on the Potsdam images due to its higher spatial resolution. Initial values of the shape parameter and the compactness parameter are typically set to 0.1 and 0.5, respectively. Table 1 lists the numbers of segmented objects for each class in the two training images.

3.2.2. OCNN structure and parameters

Considering the different spatial resolution of the two image pairs, two OCNN structures are designed: a shallow one for the Vaihingen images, and a deep one for the Potsdam images.

For the Vaihingen images, the context-guided OCNN is designed to have three parallelly stacked feature extractors (Fig. 5), where the Conv_Maxp block represents a combination of convolutional layer, activation layer, and max pooling layer. Filters for the convolutional and max pooling layers are set to 3 × 3 and 2 × 2, respectively. The initial number of convolutional filters in the first block is 32, and then is doubled in the next block. ReLU is used at the activation layer. As shown in Fig. 5, the first feature extractor containing three Conv_Maxp blocks and one SPP layer is employed for deep contextual information extraction. For deep object feature extraction, the second feature extractor consists of a shallow structure with two Conv_Maxp blocks and one SPP layer. A 3-level SPP is adopted at the end of each deep feature extractor. While a deep structure is used for the configuration of the contextual feature extractor, a shallower one is used for the object feature extractor due to the fact that the input object size is smaller than the context patch size. The third extractor receives the object deformation coefficient as input and includes a flatten layer.

The outputs from the three pipelines are concatenated through a merging layer for high-level feature fusion, followed by a dropout layer to avoid over-fitting the model. Finally, the 3-layer fully connected layers are ended with the Softmax function to generate N outputs, where N refers to the number of categories.

For the Potsdam images, the OCNN structure differs from the one displayed in Fig. 5 with regard to the number of the Conv_Maxp blocks in the two top branches. The contextual feature extractor consists of 4 Conv_Maxp blocks, while the object feature extractor of 3 Conv_Maxp blocks, because of its larger input image size and higher spatial resolution. In order to overcome the unbalanced training samples, a weight contribution with inverse class frequencies for loss correction is applied. That is, instances of small classes contribute more, whereas instances of larger classes contribute less to the final loss. In this way, losses of small classes are strongly penalized to faster the weights learning.

Other parameters are optimized empirically. Specifically, the drop rate of the dropout layer was set to 0.5, neurons of the first two fully connected layer are set to 100 and 10, respectively. Learning rate is set to 0.001. Batch sizes for the Vaihingen images and the Potsdam images in different segmentation scales are tuned according to their training samples.

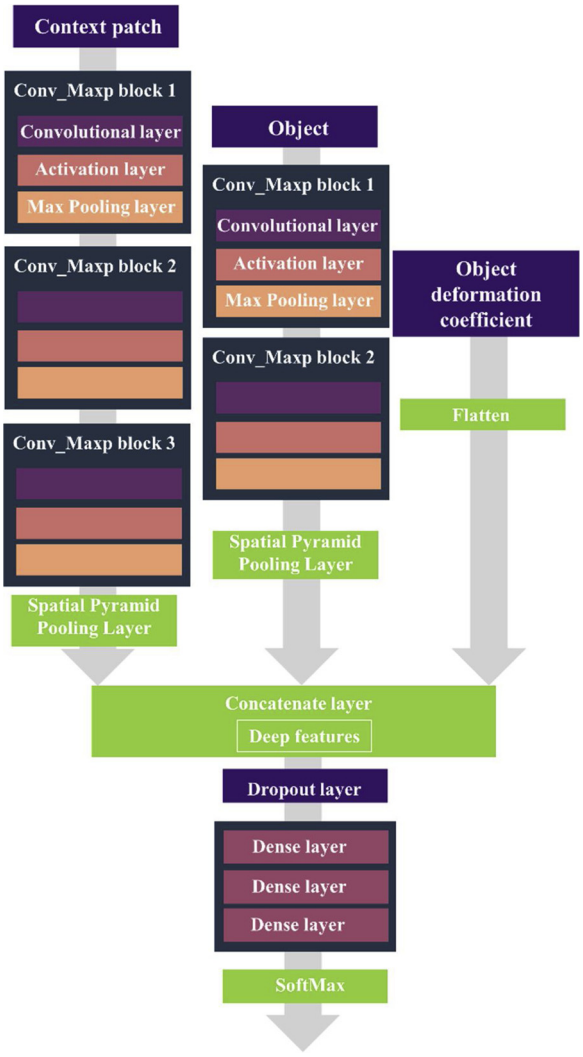

Fig. 5. Structure of the context-guided OCNN for the Vaihingen images.

3.2.3. Benchmark models and parameters

The proposed MLCG-OCNN method is compared with the following 5 benchmark methods:

1) patch-based per-pixel CNN (PBPP): PBPP utilizes a fixed-size window sliding over each pixel to generate densely overlapping patches. These patches are then fed into the CNN for per-pixel classification. A compromise between the reception field of the input image patch and the related computational cost should be considered. After testing different input sizes (i.e. 16 × 16, 32 × 32, 64 × 64, 128 × 128) 16 × 16 and 32 × 32 were selected for the Vaihingen and Potsdam images, respectively. we processed the Vaihingen images by means of a CNN with two convolutional layers and two max pooling layers, and the Potsdam images with three convolutional layers and three max pooling layers. The rest of the parameters were tuned through cross-validation;
2) patch-based per-object CNN (PBPO): Based on the per-pixel classification result, PBPO further improves the performance by masking object boundary over label pixels. The method works by counting the occurrence frequencies of each class within one object boundary and assigns, to the object, the category with the largest frequency. Firstly, the MRS algorithm was used to segment objects in the image. Segmentation parameters were set as the same as the proposed method in Section 3.2.1. Secondly, patch-based per-pixel CNNs were executed to achieve the per-pixel classification. Finally, the majority

voting was applied for per-object labeling;

3) object-based CNN (PO): Instead of feeding densely overlapping patches into the CNN, segmented objects were resized and fed into the network for discriminating objects. The benchmark OCNNs consisted solely of the object feature extractor (see Section 3.2.2). Three fully connected layers containing 100, 10, and 5 neurons were attached at the end of the feature extractor for the Vaihingen images, and the number of neurons in the last layer for the Potsdam images was configured as 6 to match with its 6 major categories. Input object sizes were optimized as 16 × 16 and 32 × 32 for the Vaihingen and Potsdam images, respectively;
4) FCN-based U-Net: U-Net (Ronneberger et al., 2015) is proven efficient in many segmentation applications, including satellite images (Rakhlin et al., 2018). We applied a standard U-Net architecture consisting of 4 downsampling layers (encoder part) and 4 upsampling layers (decoder part). 4 skip-connections were used to combine contracted features and amplified features to recover the lost information during the downsampling. To reduce the memory cost of each training iteration, images were cut into several non-overlapping sub-images with the size of 512 × 512 pixels before being fed into the U-Net. Results of the sub-images were merged for performance assessment. Other parameters were tuned empirically by referring to the literature;
5) state-of-the-art semantic segmentation architecture DeepLabV3+ (DLV + 3): DeepLabV3+ is a state-of-the-art semantic segmentation architecture proposed by Google (Chen et al., 2018a, 2018b). DeepLabV3+ outperforms its predecessors by absorbing a set of key advanced designs including the Atrous Convolution layer, the Atrous Spatial Pyramid Pooling layer, and Skip-connections. In this paper, we used Resnet101 as the backbone of the network. Parameter settings of the DeepLabV3+ were configured according to the open source repository released by Chen et al. (2019a, 2019b).

3.3. Results and discussion

3.3.1. Segmentation scale parameter (SSP)

SSP affects the classification performance by controlling the size of segmented objects. As mentioned in Section 3.2.1, from the Vaihingen dataset the SSP value of 20 generates 33726 objects Table 1), each of them containing averagely 214 pixels. These segmented objects are then resized into 16 × 16 (256 pixels). Using the SSP value of 50, 7430 objects were generated from the Vaihingen training dataset. The average number of pixels for objects is 970, which is nearly equal to a 32 × 32 square with 1024 pixels. Therefore, segmented objects were resized into 32 × 32. Similarly, segmented objects from the Potsdam dataset were resized into 32 × 32 and 72 × 72, respectively, using the SPP values of 40 and 100. The sizes of context patches for the two datasets were set as twice as the size of objects, and the S_{re} in Eq. (3) was set to 2.0. Pixel-based overall accuracy (OA) and Kappa coefficient (kappa) were used to assess the results. Fig. 6 shows the results of classification using the SSPs on the two testing images. With regard to the Vaihingen image, moving from the ground reference (Fig. 6a) to the results obtained with SSP 50, CRF (Fig. 6e), we see that classification results contain less pixels classified as low vegetation and more pixels classified as tree. In the Potsdam image, moving from the ground reference (Fig. 6f) to the results achieved with SSP 100, CRF (Fig. 6j), we see that the last two maps identify trees within the major central buildings better than the SSP 40 non-CRF (Fig. 6g) and 40, CRF (Fig. 6h). Results of SSP 40 non-CRF and 40, CRF are better in identifying the surrounding buildings than results in SSP 100 non-CRF (Fig. 6i) and SSP 100, CRF (Fig. 6j).

Table 2 shows the results of our quantitative assessment. Considering the Vaihingen image, the image using SSP of 20 outperformed the one using SSP of 50 by an increase of 1.78 % in OA. The OA of the image using the SSP value of 20 increased 3.06 % after using the CRF refinement. CRF led to an improvement of 2.24 % on the image using the SSP 50. Similarly, for the Potsdam image, the SSP 40 outperformed the SSP 100 by an increase of 6.12 % in OA, and the gap was then enlarged to 7.27 % after the CRF refinement. The CRF refinement led to OA improvement by 1.28 % and 0.13 %, respectively, on the two SSPs for the Potsdam image. This demonstrates that a low segmentation scale level (i.e., low SSP) can present a better performance on per-object classification. Moreover, the boost effect owing to the CRF refinement was more significant on classification results at a low segmentation scale. The phenomenon can be explained as follows: 1) the number of mismatched pixels between the reference image and the segmented image increases as the SSP becomes higher; 2) when the segmentation scale becomes higher, small objects are more likely to be absorbed by large objects in different classes. Thus, features belonging to different classes are mixed and the classification performance decreases.

3.3.2. Context patch size

Following Eq. (3), to assess the effects of the context patch size, the parameter S_{re} is set empirically to values of 2.0, 2.5, 3.0, 3.5, 4.0, 5.0, where these different values help to perform the comparative experiment. Images using a low SSP can perform better (see Section 4.3.1). Table 3 shows the results on the Vaihingen and Potsdam testing images segmented by the SSP of 20 and 40, respectively, and using different S_{re} values.

The Vaihingen image with a S_{re} equal to 2.5 achieved the highest OA 78.93 % with a Kappa coefficient of 0.706. The OA and the Kappa coefficient dropped slightly as S_{re} increases. In the Potsdam image, the highest OA 80.08 % and Kappa coefficient 0.694 were achieved by a S_{re} equal to 4.0. The result does not show a distinct correlation between S_{re} and the classification performance.

Although it is not clear how each S_{re} influences the performance of the OCNN, the fluctuation of the OA and Kappa coefficient for both testing images at different S_{re} values is small. Therefore, the size of context patches may not be a deterministic parameter for the classification performance of the OCNN. However, with regard to the computational efficiency, a small S_{re} is recommended, because it involves a low volume of input pixels in computation.

3.3.3. Comparison with benchmark methods

MLCG-OCNN was compared with the existing benchmark methods (see Sections 4.2.3). Segmentation scales for the Vaihingen and Potsdam datasets were set at 20 and 40, respectively. The size of context patches were tuned twice the object size considering the trade- off between the computational cost and classification performance. Classification results are discussed based on visual interpretation and qualitative assessment of the output classification maps by means of pixel-based OA, kappa coefficient (Kappa), as well as per-class precision (Pre) and recall (Recall).

3.3.3.1. Classification performance: the Vaihingen image. Fig. 7 presents the classification results for visual interpretation. MLCG-OCNN achieved a good result with high internal compactness and fine boundary delineation. In particular, MLCG-OCNN outperformed the other 5 benchmark methods when classifying objects with straight edges and sharp corners such as roads and buildings. The pixel-wise method PBPP demonstrated capabilities for high-level feature extraction, e.g., most roads and buildings were correctly discriminated. Yet the recall of the car category by PBPP was very low. Compared with PBPP, object boundaries were refined by PBPO. Moreover, isolated and misclassified pixels within objects on PBPP results were rectified by PBPO. MLCG-OCNN also demonstrated a strong ability to eliminate isolated and misclassified pixels. Moreover, the recall of the car category on MLCG-OCNN improved significantly in comparison with PBPP and PBPO. The result of PO was the worst among all the methods. Semantically consecutive objects were shattered into pieces and assigned to different classes due to the over-segmented images. Although over-segmented images were used in

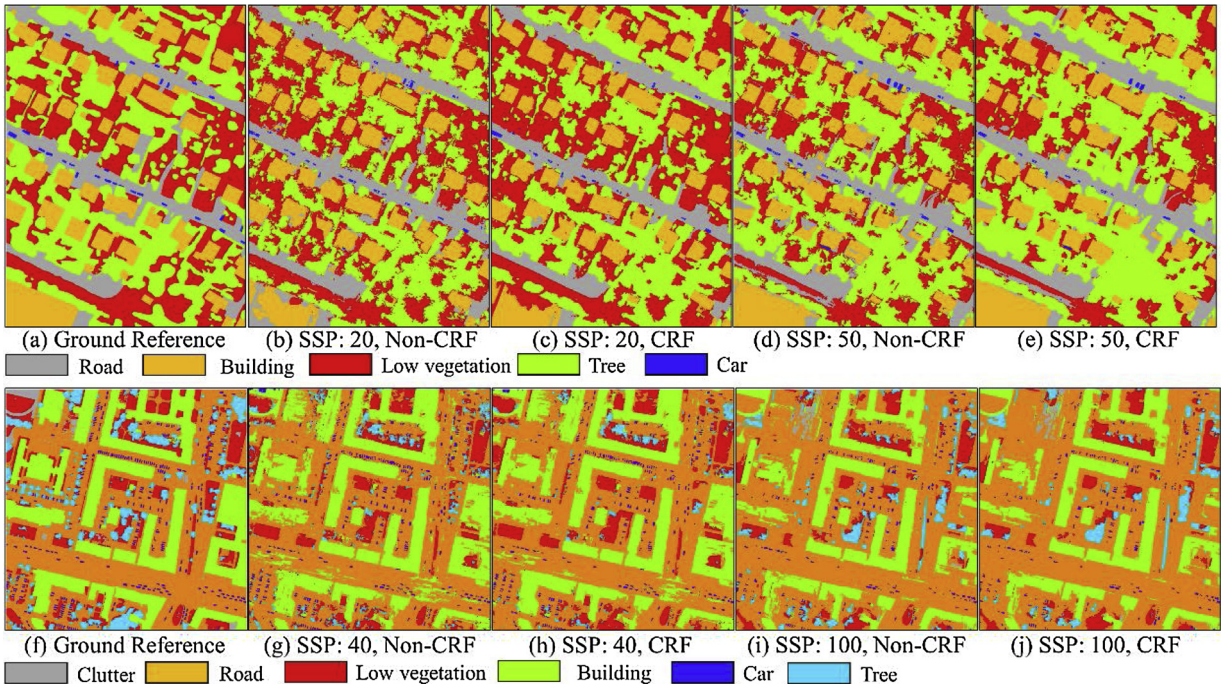

Fig. 6. Classification results using different SSPs: (a-e) the Vaihingen image, and (f–j) the Potsdam image.

Table 2
Overall accuracy (OA) and Kappa coefficient (Kappa) for the two testing images with different SSPs.

	SSP	CRF	OA	Kappa
Vaihingen	20	no	77.70	0.697
		yes	80.76	0.732
	50	no	75.92	0.667
		yes	78.16	0.694
Potsdam	40	no	79.01	0.676
		yes	80.29	0.693
	100	no	72.89	0.569
		yes	73.02	0.566

Table 3
Overall accuracy (OA) and Kappa coefficient (Kappa) for the two testing images in different S_{re} values.

	S_{re}	2.0	2.5	3.0	3.5	4.0	5.0
Vaihingen	OA	77.13	78.93	78.57	78.22	78.30	77.25
	Kappa	0.683	0.706	0.702	0.700	0.700	0.683
Potsdam	OA	79.01	78.63	77.65	79.13	80.08	77.56
	Kappa	0.676	0.672	0.653	0.672	0.694	0.650

MLCG-OCNN, much better classification performance was achieved compared to PO, because multi-level context information was incorporated. For FCN based methods, U-Net and DLV3+ produced maps with smooth boundaries and high internal compactness. Furthermore, U-Net and DLV3+ were more capable to distinguish the car category compared to PBPP and PBPO. In this regard, MLCG-OCNN provides similar results as the U-Net. However, U-Net and DLV3+ tend to smooth the straight edges and round the sharp corners on the road and building classes (Fig. 7).

Table 4 provides the quantitative assessment of the classification performance. MLCG-OCNN achieved the highest OA 81.03 % with Kappa 0.734, compared with PBPP (OA 75.94 % and Kappa 0.664), PO (OA 64 % and Kappa 0.501), PBPO (OA 76.17 % and Kappa 0.667), U-Net (OA 78.30 % and Kappa 0.697), and DLV3+ (OA 80.71 % and Kappa 0.730). The advantages of MLCG-OCNN can be observed obviously among the conventional pixel-wise and object-wise methods. Even compared with the state-of-the-art architecture DeepLabV3+ (i.e. DLV3+), the MLCG-OCNN method gained an OA improvement of 0.32 % and a kappa coefficient improvement of 0.41. Therefore, we can conclude that the accuracies achieved with DeepLabV3+ and MLCG-OCNN are comparable.

In terms of classification performance on each class, high precisions and recalls were achieved by MLCG-OCNN. Compared with PBPP, PO, PBPO, and U-Net, MLCG-OCNN gained significant increases on precisions of road, building, and car classes. In particular, when compared with the object-based CNN method, PO, precisions of road and building classes increased dramatically by 30.57 % and 7.66 %, respectively. Although precisions of road and building classes achieved with PBPO are close to MLCG-OCNN, their recalls are low. Moreover, precisions of the low vegetation and car classes were much lower than the proposed method by a decrease of 10 % and 40.7 %, respectively. For the semantic segmentation methods, the performance of DLV3+ on different classes increased slightly compared with U-Net owing to its complex and deeper architecture. In addition, DLV3+ demonstrated its advantage in dealing with unbalanced data by achieving a precision of 90.68 % and a recall of 47.46 % on car class, which is higher than the results of the other methods.

3.3.3.2. Classification performance: the Potsdam image. The Potsdam images have a higher spatial resolution, much more artificial objects showing complex spectral variances and small objects augmenting the complexity of the major classes. Moreover, the image quality of the Potsdam images is not guaranteed. Straight outlines of buildings were twisted into irregular curves due to noise disturbance. Since most leaves of trees were already fallen when the images were collected, features below the sparse branches such as roads and cars were mixed up with tree features. This greatly hampered the distinction of the tree class. Moving cars result in image ghosting and tearing. These issues increased significantly the complexity of the classification, especially for the object-based classification method.

As shown in Fig. 8, results with performances similar to the Vaihingen image were obtained from MLCG-OCNN and the other benchmark methods, except for the PO. Due to the high inter and intra-class

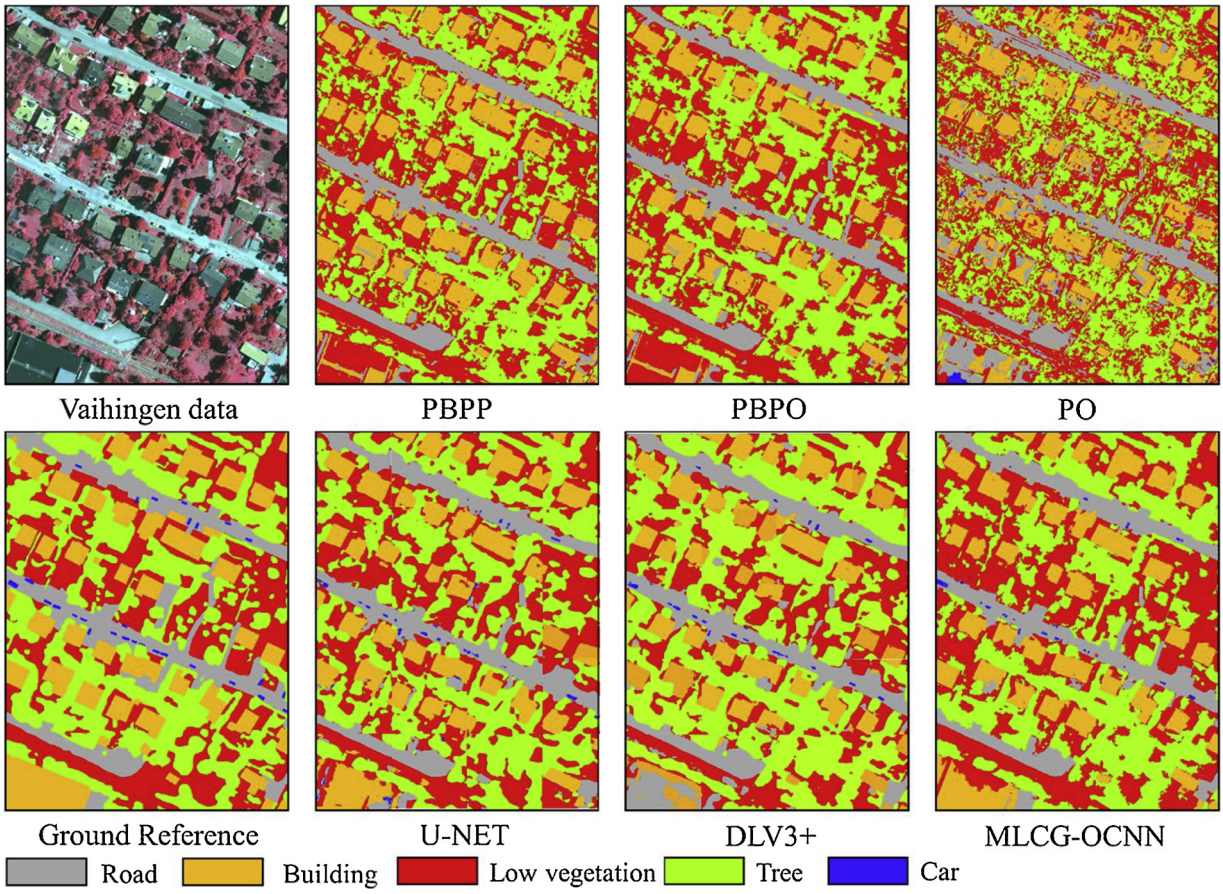

Fig. 7. Comparison of the classification results achieved with MLCG-OCNN and the other five benchmark methods on the Vaihingen image.

Table 4
Quantitative assessment of classification performance on the Vaihingen image using the precision, recall, Overall Accuracy (OA), and Kappa.

	Class	PBPP	PO	PBPO	U-NET	DLV3+	MLCG-OCNN
Precision	Tree	80.79	69.84	80.57	80.34	78.86	79.63
Recall		77.88	67.35	78.64	80.96	90.79	85.99
Precision	Low	60.57	50.49	60.79	66.88	77.41	70.79
Recall	vegetation	73.10	57.40	73.56	72.43	66.95	72.50
Precision	Road	88.91	60.00	90.30	85.70	76.42	90.57
Recall		73.76	70.21	72.86	78.98	82.56	80.12
Precision	Building	88.44	87.09	89.50	89.79	94.39	94.75
Recall		79.96	64.29	79.62	82.65	81.58	86.35
Precision	Car	37.38	17.65	47.52	42.85	90.68	88.22
Recall		79.54	10.57	71.06	36.99	47.46	32.08
OA		75.94	64.00	76.17	78.30	80.71	81.03
Kappa		0.664	0.501	0.667	0.697	0.730	0.734

spectral variations under the super high resolution, over-segmented objects exhibited complex patterns on high-level features, which resulted in the poor performance of the per-object classification method (PO). Similar to the experiments on the Vaihingen image, the pixel-wise method PBPP on the Potsdam image also suffered the salt-and-pepper effect. In contrast, DLV3+ produced a result with the highest object integrity than other methods. Although slightly inferior to the DLV3+ in terms of the object integrity, MLCG-OCNN achieved a desirable result with major objects maintaining fine boundaries. Most roads, buildings, and low vegetation were accurately discriminated with straight edges.

MLCG-OCNN achieved an OA 80.29 % and a Kappa 0.693, which are larger than the other methods, PBPP (OA 78.4 % and Kappa 0.667), PO (OA 67.45 % and Kappa 0.483), PBPO (OA 80.05 % and Kappa 0.691), and U-Net (OA 78.37 % and Kappa 0.743) (Table 5). Different from the test on the Vaihingen image, DLV3+ demonstrated its superiority by achieving OA 83.54 % and Kappa 0.743. DLV3+ provided the most accurate result with respect to each class. Compared with PO, MLCG-OCNN achieved remarkable precision improvement on each class with the highest value equal to 27.62 %. When compared with PBPP, MLCG-OCNN achieved moderate increases of per-class precision with the values 0.64 %, 4.64 %, 8.76 %, and 10.38 % on road, building, car, and clutter classes respectively. The performance gaps on road and building classes were further narrowed between PBPO and MLCG-OCNN. Considering building, tree, and low vegetation classes, PBPO showed slightly better result.

3.3.3.3. Computational efficiency. Because deep learning techniques have computational cost of training complex networks and require significant volumes of training data, we assessed not only the classification performance, but also the computational efficiency. We applied three indicators: the volume of training samples, time consumption on each phase, and the number of network parameters. Time consumption of each method was decomposed into 3 main elements: the running time of data loading, the time consumption of the model training phase and the duration of the model inference phase.

Tests were implemented on a machine with a NVIDIA P5000 GPU and 64 GB memory. Table 6 shows the comparison of computational efficiency among the methods on the Vaihingen image. PO and U-NET spent almost the same time on the model training and inference phases, with a total time near to 9 min. Due to the resize operation in the data preprocessing, PO consumed nearly one minute in the data loading phase. PBPP has a relatively small number of model parameters, and spent the least time (4 min) in the model training phase. Whereas in the model inference phase, the densely overlapping patches covering all

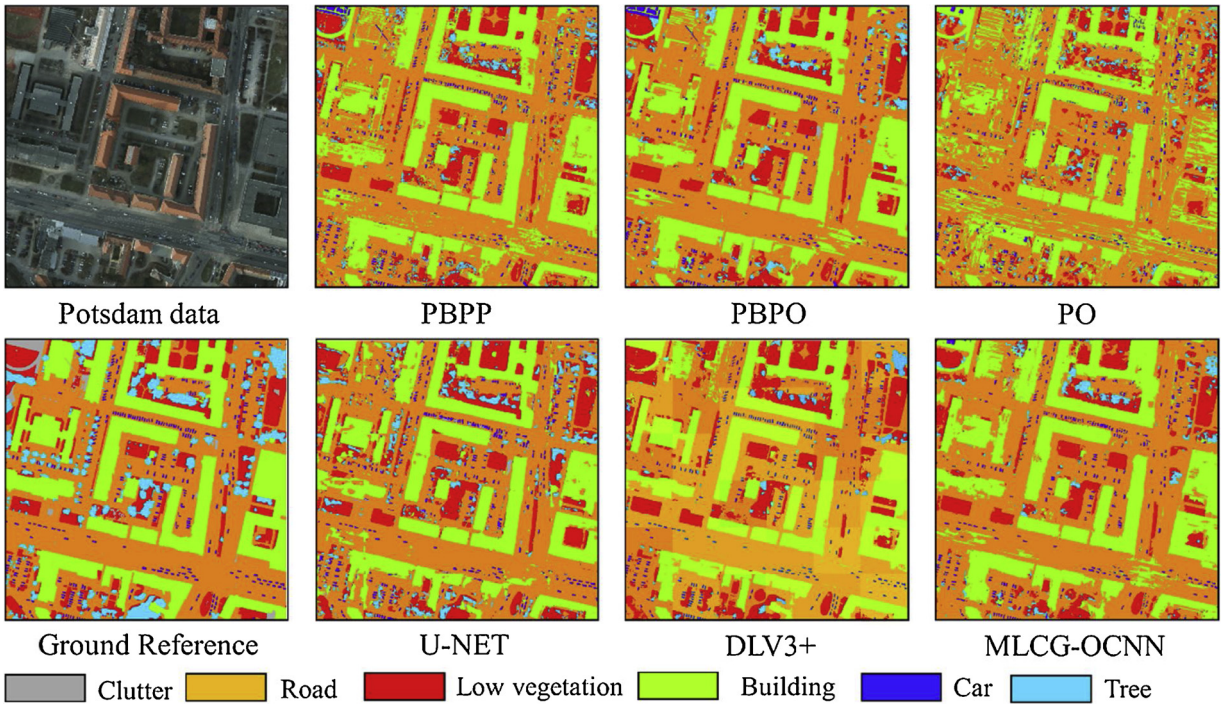

Fig. 8. Classification results of the MLCG-OCNN method and the other five benchmark methods on the Potsdam image.

Table 5
Quantitative assessment of the classification accuracy on the Potsdam image using the precision, recall, Overall Accuracy (OA), and Kappa.

	Class	PBPP	PO	PBPO	U-NET	DLV3+	MLCG-OCNN
Precision	Clutter	15.04	10.88	9.39	9.70	77.72	25.42
Recall		3.62	1.59	4.07	6.99	2.02	1.66
Precision	Road	79.87	68.19	80.18	83.01	82.16	80.51
Recall		89.23	86.02	92.04	87.87	97.32	92.89
Precision	Building	80.55	71.53	85.77	88.65	94.62	85.19
Recall		88.33	70.51	89.31	85.78	88.79	92.25
Precision	Tree	72.10	43.01	73.71	53.47	97.27	70.63
Recall		24.02	13.69	22.24	31.85	18.99	19.72
Precision	Car	71.13	55.47	68.67	68.90	92.21	79.89
Recall		69.99	55.43	64.16	67.95	68.44	67.04
Precision	Low	70.30	60.40	72.10	58.38	64.95	67.65
Recall	vegetation	62.14	37.11	65.08	68.58	75.53	58.50
OA		78.40	67.45	80.05	78.37	83.45	80.29
Kappa		0.667	0.483	0.691	0.676	0.743	0.693

Table 6
Comparison of computational efficiency on the Vaihingen image. (Mins is abbreviation of minutes).

Method	Training samples	Data loading (Mins)	Model training (Mins)	Model inference (Mins)	Model parameters
PBPP	112,640	0.1	4	12	226,457
PO	33,726	1	6	2	197,913
PBPO	112,640	0.1	4	12	226,457
U-Net	25	0.1	7	0.05	7,768,198
DLV3+	25	0.1	90	0.2	118,821,805
MLCG-OCNN	33,726	2	12	4	345,877

pixels in the image were clipped and fed into the model for per-pixel classification, and thus the highly redundant data increased significantly the time consumption, up to 12 min. Since PBPO predicted the image based on the classification result of the PBPP, the computational efficiency was the same as the PBPP. DLV3+ spent the longest time of 1.5 h on the model training phase due to its deep complex architecture with the enormous amount of model parameters up to 118,821,805. In contrast, MLCG-OCNN spent only 12 min on the model training phase and 18 min in total, which is 7.5 and 5 times faster than DeepLabV3+, respectively.

Table 7 shows the comparison of computational efficiency among the different methods on the Potsdam image. Again, the object-based method PO achieved the best computational efficiency with a total running time of 10.8 min. For PBPP, as the number of predicted pixels increased from 1921 × 2574 (the Vaihingen image) to 6000 × 6000 (the Potsdam image), the time consumption in the model inference phase increased dramatically to 120 min. Meanwhile, the running time during the training stage also increased to 20 min due to the deeper network operating with a doubled volume of training parameters. Although the number of parameters of U-Net was 15 times larger than the patch-based methods (PBPP and PBPO), the running time of the model inference was reduced greatly to 0.08 min, similar to the case (0.05 min), because the fully convolutional networks took an image as model input. DLV3+ took the longest time during the model training (i.e. 120 min) to reach the desired performance. Comparatively, MLCG-OCNN lasted in total no more than 30 min, of which 1.8, 21 and 7 min for data loading, model training, and model inference, respectively. In terms of the overall time consumption, MLCG-OCNN was 4 times faster than DeepLabV3 + .

As shown in Table 5 benefiting from the large volume of training

Table 7
Comparison of computational efficiency on the Potsdam image.

Method	Training samples	Data loading (Mins)	Model training (Mins)	Model inference (Mins)	Model parameters
PBPP	140,625	0.05	20	120	505,124
PO	21,024	0.8	6	4	735,524
PBPO	140,625	0.05	20	120	505,124
U-Net	36	0.1	22	0.08	7,768,198
DLV3+	36	0.1	120	0.5	118,821,805
MLCG-OCNN	21,024	1.8	21	7	591,072

samples in the Potsdam dataset, the pixel-wise CNN demonstrates its strong model generalization ability on the large dataset. However, two drawbacks of the pixel-wise methods can be observed from the experiments. Firstly, their result suffers from the salt-and-pepper effect, with blurred boundaries occurring between inter-class objects due to loss of object level features. Secondly, the prediction of the densely overlapping patches is quite time consuming (Tables 6 and 7). Therefore, the object-based CNN has been proposed. Rather than feeding fixed size patches into the CNN, OCNN feeds directly segmented objects into the network. However, the poor performance of the per-object method (PO) in Tables 4 and 5 shows that this method is still not sufficient to use solely the independent object features for discriminating objects. In the paper, the proposed feature fusion network accepting multi-inputs is proposed to discriminates objects by learning simultaneously object features and their contextual information. The boundaries of segmented objects are used for object mask in order to maintain shape information of objects. To solve the issue of missing scale information during the resize operation, the object deformation coefficient is proposed and incorporated into the network. The contextual guidance of the method is decomposed into two levels: the object-level and the pixel-level. The object-level context patch is object-dependent, the size of which is determined by the object size. In this way, the feature-fusing network achieves an accurate per-object classification result while maintaining fine object boundaries. The pixel level contextual guidance, with the assistance of the CRF, is used furthermore, to improve the classification performance at the pixel level.

The experimental results on the two image pairs from different datasets demonstrate the effectiveness of the proposed method. Specifically, for the image pair with a small number of training samples (i.e. Vaihingen), the method achieves the best result with OA and Kappa exceeding benchmark methods (Table 4). This can greatly benefit the classification in practical scenarios where there are only a limited number of available training samples. For the Potsdam image pair with a relatively large number of training samples, the gap of the classification performance among the methods was smaller than that on the Vaihingen image pair due to its larger training samples (Table 5). MLCG-OCNN outperformed the pixel-wise PBPP, the object-wise PBPO and PO, as well as the pixel-to-pixel U-Net. Compared with the latest semantic segmentation method, i.e. DeepLabV3+ from the computer vision domain, the method in this paper has a slightly lower performance, with OA decrease by 3.16 % and Kappa by 0.05. However, the time consumption on the model training stage for the DeepLabV3+, due to its deep architecture and massive parameters, was 5 times longer than that in MLCG-OCNN (Table 7). The traditional pixel-wise CNN (i.e. PBPP and PBPO) is the most time-consuming on the model inference phase due to the prediction of densely overlapping patches. In contrast, MLCG-OCNN reduced remarkably the running time by moving from per-pixel prediction to per-object prediction. Therefore, the training time of the proposed feature fusion network is much more acceptable with less compromise to the classification performance.

It should be noted that the comparison experiments were performed over two image pairs from the ISPRS datasets. Thus, the overall accuracy and the Kappa coefficient reported in the experiments are lower than the state of art results that used all images in the ISPRS datasets for model training. The motivation for using small datasets in the experiment is that there are limited annotated images available for model training in practical scenarios. Manual image labelling is quite labor-intensive. For example, it takes a remote sensing expert about 2 h to manually label 25 images (each containing 512*512 pixels). If summed with the time required for model training, the total time consumption can be very high in practical usages. It has been acknowledged that deep learning architectures are highly dependent on training datasets. If the training dataset is too small, the model may encounter an overfitting problem. While if the dataset is too large, time-consumption on model training can be very high. Therefore, a compromise between classification performance and computational efficiency has to be made. In this situation, it is critical for models to maintain their robustness on small datasets. This has not been tested in the cited studies, therefore the present paper provides a novel contribution in this regard. Meanwhile, deep learning with small datasets has also raised interests from both deep learning and remote sensing societies in recent years (Liu and Deng, 2016; Pasupa and Sunhem, 2017; Zhao, 2017; Zhu et al., 2019; Mishra et al., 2019; Wu et al., 2019a, 2019b). In this paper, on one hand, the proposed object-based classification method is performed over datasets containing more than 30,000 objects. i.e., a suitable data volume to avoid model overfitting. On the other hand, we intentionally explore the classification performance and computational efficiency of our method and benchmark methods on small datasets, with an effort to validate model robustness on small datasets, as well as to provide potential guidance for practical usages. Existing FCN-based methods e.g., U-Net and DeepLabV3+ require that each pixel is labeled. MLCG-OCNN instead does not require the training image to be fully labeled, because it is trained over segmented objects. Labeled objects can be clipped from the partially labeled images and thus are fed into the OCNN for classification. When only few images are available, the work provides a valuable reference on how benchmark methods perform, compared with the proposed method.

4. Conclusions and future work

To deal with the known challenges of VHRI classification, MLCG-OCNN is proposed as an effective method with high computational efficiency. The MLCG-OCNN method consists of an object-level contextual guided object-based CNN and is applied to carry out per-object classification by fusing the high-level features of spectral patterns, geometric characteristics, and contextual information. Then, with the help of the CRF, the per-object classification result is further refined by means of the pixel-level contextual guidance. The method is compared with 5 benchmark methods including the state-of-the-art network DeepLabV3 + . The experimental results achieved by processing images from different datasets demonstrate that the method has a remarkable performance for VHRI classification, especially when it is utilized on small datasets. Moreover, it shows a high computational efficiency on both the model training and inference stages.

MLCG-OCNN contributes to OCNN (currently gaining increasing attention in the remote sensing society) as follows:

(1) The object contour-preserving mask strategy with the supplement of object deformation coefficient to complement the high-level feature extraction. The work suggests a combination of spectral features with geometric characteristics to discriminate objects.
(2) The incorporation of independent object features with multi-level contexts to improve VHRI classification. Firstly, the object level context is used to guide the per-object classification by the OCNN. Secondly, the pixel-level context is used to refine the classification result at the pixel level.
(3) The high computational efficiency with competitive classification performance. On the one hand, the object-as-input network avoids the time-consuming pixel-wise operation. On the other hand, the comparatively simple network of moderate parameter volume saves plenty of time on model training.

It should be noted that the object deformation coefficient might be insufficient to characterize the object geometric characteristics. Therefore, further investigations on new features measuring geometric characteristics are needed to improve the results.

In addition, considering segmented objects in very small sizes, on the one hand, their geometric characteristics may not be sufficient to allow object discrimination due to their irregular shapes, and on the other hand, small objects and the corresponding context patches contain small number of pixels. These two obstacles make it difficult for OCNN to mine representative deep features for discriminating small

objects. Therefore, the fusion strategy of merging very small objects into semantic big objects will be studied to improve the classification performance on small objects.

Finally, the experiment was intentionally performed over small datasets with limited training samples. In future work, the full ISPRS image datasets can be considered.

Author statement

Chenxiao Zhang: Conceptualization, Methodology, Software, Validation, Investigation. **Peng Yue:** Writing- Original draft preparation, Writing – Review & Editing, Supervision. **Deodato Tapete:** Writing- Original draft preparation, Writing – Review & Editing. **Boyi Shangguan:** Data curation. **Mi Wang:** Resources. **Zhaoyan Wu:** Software, Validation, Investigation.

Declaration of Competing Interest

Each of the authors confirms that no part of this manuscript has been previously published, nor is any part is currently under consideration by any other journal. Additionally, each of the authors has approved the contents of this paper and have agreed to the submission policies of International Journal of Applied Earth Observation and Geoinformation.

Acknowledgements

We appreciate the reviewers and editors for their constructive comments that helped improve the quality of the paper. The work was supported by Major State Research Development Program of China (No. 2017YFB0504103), National Natural Science Foundation of China (No. 41722109, 61825103, 91738302), Hubei Provincial Natural Science Foundation of China (No. 2018CFA053), and Wuhan Yellow Crane Talents (Science) Program (2016).

References

Agarap, A.F., 2017. An Architecture Combining Convolutional Neural Network (CNN) and Support Vector Machine (SVM) for Image Classification. arXiv preprint arXiv:1712.03541. .

Alhichri, H., Alajlan, N., Bazi, Y., Rabczuk, T., 2018. Multi-scale convolutional neural network for remote sensing scene classification. 2018 IEEE International Conference on Electro/Information Technology (EIT) 113–117. https://doi.org/10.1109/EIT.2018.8500107.

Baatz, M., Schape, A., 2000. Multiresolution segmentation - an optimization approach for high quality multi-scale image segmentation. Beutrage Zum AGIT-Symposium 12–23.

Castelluccio, M., Poggi, G., Sansone, C., Verdoliva, L., 2015. Land Use Classification in Remote Sensing Images by Convolutional Neural Networks. arXiv preprint arXiv:1508.00092. .

Chen, L.-C., Zhu, Y., Papandreou, G., Hui, H., 2018a. DeepLab: Deep Labelling for Semantic Image Segmentation. (Accessed 15 January 2020). https://github.com/tensorflow/models/tree/master/research/deeplab.

Chen, L.C., Zhu, Y., Papandreou, G., Schroff, F., Adam, H., 2018b. Encoder-decoder with atrous separable convolution for semantic image segmentation. European Conference on Computer Vision (ECCV) 833–851. https://doi.org/10.1007/978-3-030-01234-2_49.

Chen, C., Gong, W., Chen, Y., Li, W., 2019a. Learning a two-stage CNN model for multi-sized building detection in remote sensing images. Remote Sens. Lett. 10, 103–110. https://doi.org/10.1080/2150704X.2018.1528398.

Chen, Y., Ming, D., Lv, X., 2019b. Superpixel based land cover classification of VHR satellite image combining multi-scale CNN and scale parameter estimation. Earth Sci. Inform. 1–23. https://doi.org/10.1007/s12145-019-00383-2.

Cheng, G., Zhou, P., Han, J., 2016. Learning rotation-invariant convolutional neural networks for object detection in VHR optical remote sensing images. IEEE Trans. Geosci. Remote Sens. 54, 7405–7415. https://doi.org/10.1109/TGRS.2016.2601622.

Fu, T., Ma, L., Li, M., Johnson, B.A., 2018. Using convolutional neural network to identify irregular segmentation objects from very high-resolution remote sensing imagery. J. Appl. Remote Sens. 12, 1. https://doi.org/10.1117/1.jrs.12.025010.

Hu, F., Xia, G.S., Hu, J., Zhang, L., 2015. Transferring deep convolutional neural networks for the scene classification of high-resolution remote sensing imagery. Remote Sens. 7, 14680–14707. https://doi.org/10.3390/rs71114680.

Ichoku, C., Karnieli, A., 1996. A review of mixture modeling techniques for sub-pixel land cover estimation. Remote Sens. Rev. 13, 161–186. https://doi.org/10.1080/02757259609532303.

[dataset] ISPRS, 2013a. ISPRS 2D Semantic Labeling–Vaihingen data. http://www2.isprs.org/commissions/comm3/wg4/2d-sem-label-vaihingen.html.

[dataset] ISPRS, 2013b. ISPRS 2D Semantic Labeling–Potsdam data. http://www2.isprs.org/commissions/comm3/wg4/2d-sem-label-potsdam.html.

Lecun, Y., Bengio, Y., Hinton, G., 2015. Deep learning. Nature. https://doi.org/10.1038/nature14539.

Liu, S., Deng, W., 2016. Very deep convolutional neural network based image classification using small training sample size. 2015 3rd IAPR Asian Conference on Pattern Recognition (ACPR) 730–734. https://doi.org/10.1109/ACPR.2015.7486599.

Luus, F.P.S., Salmon, B.P., Van Den Bergh, F., Maharaj, B.T.J., 2015. Multiview deep learning for land-use classification. IEEE Geosci. Remote Sens. Lett. 12, 2448–2452. https://doi.org/10.1109/LGRS.2015.2483680.

Lv, X., Ming, D., Lu, T., Zhou, K., Wang, M., Bao, H., 2018. A new method for region-based majority voting CNNs for very high resolution image classification. Remote Sens. 10, 1946. https://doi.org/10.3390/rs10121946.

Lv, X., Ming, D., Chen, Y.Y., Wang, M., 2019. Very high resolution remote sensing image classification with SEEDS-CNN and scale effect analysis for superpixel CNN classification. Int. J. Remote Sens. 40, 506–531. https://doi.org/10.1080/01431161.2018.1513666.

Marmanis, D., Datcu, M., Esch, T., Stilla, U., 2016. Deep learning earth observation classification using ImageNet pretrained networks. IEEE Geosci. Remote Sens. Lett. 13, 105–109. https://doi.org/10.1109/LGRS.2015.2499239.

Mishra, S., Yamasaki, T., Imaizumi, H., 2019. Improving Image Classifiers for Small Datasets by Learning Rate Adaptations. arXiv preprint arXiv:1903.10726. .

Neubert, M., Herold, H., Meinel, G., 2008. Assessing image segmentation quality–concepts, methods and application. Object-based Image Analysis. pp. 769–784. https://doi.org/10.1007/978-3-540-77058-9_42.

Nguyen, T., Han, J., Park, D.C., 2013. Satellite image classification using convolutional learning. AIP Conf. Proc. 1558, 2237–2240. https://doi.org/10.1063/1.4825984.

Othman, E., Bazi, Y., Alajlan, N., Alhichri, H., Melgani, F., 2016. Using convolutional features and a sparse autoencoder for land-use scene classification. Int. J. Remote Sens. 37, 2149–2167. https://doi.org/10.1080/01431161.2016.1171928.

Pasupa, K., Sunhem, W., 2017. A comparison between shallow and deep architecture classifiers on small dataset. 2016 8th International Conference on Information Technology and Electrical Engineering (ICITEE) 1–6. https://doi.org/10.1109/ICITEED.2016.7863293.

Rakhlin, A., Davydow, A., Nikolenko, S., 2018. Land cover classification from satellite imagery with U-net and lovasz-Softmax loss. CVPR Workshops 262–266. https://doi.org/10.1109/CVPRW.2018.00048.

Ren, X., Guo, H., Li, S., Wang, S., Li, J., 2017. A novel image classification method with CNN-XGBoost model. International Workshop on Digital Watermarking 378–390. https://doi.org/10.1007/978-3-319-64185-0_28.

Richmond, D.L., Kainmueller, D., Yang, M.Y., Myers, E.W., Rother, C., 2015. Relating Cascaded Random Forests to Deep Convolutional Neural Networks for Semantic Segmentation. arXiv preprint arXiv:1507.07583. .

Ronneberger, O., Fischer, P., Brox, T., 2015. U-net: convolutional networks for biomedical image segmentation. Int. Conf. Medical Image Comput. Comput.-Assisted Intervention 234–241. https://doi.org/10.1007/978-3-319-24574-4_28.

Sharma, A., Liu, X., Yang, X., Shi, D., 2017. A patch-based convolutional neural network for remote sensing image classification. Neural Netw. 95, 19–28. https://doi.org/10.1016/j.neunet.2017.07.017.

Van den Bergh, M., Boix, X., Roig, G., Van Gool, L., 2015. Seeds: superpixels extracted via energy-driven sampling. Int. J. Comput. Vis. 111, 298–314. https://doi.org/10.1007/s11263-014-0744-2.

Vetrivel, A., Gerke, M., Kerle, N., Nex, F., Vosselman, G., 2018. Disaster damage detection through synergistic use of deep learning and 3D point cloud features derived from very high resolution oblique aerial images, and multiple-kernel-learning. ISPRS J. Photogramm. Remote Sens. 140, 45–59. https://doi.org/10.1016/j.isprsjprs.2017.03.001.

Weng, Q., Mao, Z., Lin, J., Guo, W., 2017. Land-use classification via extreme learning classifier based on deep convolutional features. IEEE Geosci. Remote Sens. Lett. 14, 704–708. https://doi.org/10.1109/LGRS.2017.2672643.

Wu, M., Zhao, X., Sun, Z., Guo, H., 2019a. A hierarchical multiscale super-pixel-based classification method for extracting urban impervious surface using deep residual network from WorldView-2 and LiDAR data. IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens. 12, 210–222. https://doi.org/10.1109/JSTARS.2018.2886288.

Wu, W., Li, H., Li, X., Guo, H., Zhang, L., 2019b. PolSAR image semantic segmentation based on deep transfer learning - realizing smooth classification with small training sets. IEEE Geosci. Remote Sens. 16, 977–981. https://doi.org/10.1109/LGRS.2018.2886559.

Zhang, P., Gong, M., Su, L., Liu, J., Li, Z., 2016. Change detection based on deep feature representation and mapping transformation for multi-spatial-resolution remote sensing images. ISPRS J. Photogramm. Remote Sens. 116, 24–41. https://doi.org/10.1016/j.isprsjprs.2016.02.013.

Zhang, Ce, Pan, X., Li, H., Gardiner, A., Sargent, I., Hare, J., Atkinson, P.M., 2018a. A hybrid MLP-CNN classifier for very fine resolution remotely sensed image classification. ISPRS J. Photogramm. Remote Sens. 140, 133–144. https://doi.org/10.1016/j.isprsjprs.2017.07.014.

Zhang, Ce, Sargent, I., Pan, X., Li, H., Gardiner, A., Hare, J., Atkinson, P.M., 2018b. An object-based convolutional neural network (OCNN) for urban land use classification. Remote Sens. Environ. 216, 57–70. https://doi.org/10.1016/j.rse.2018.06.034.

Zhang, Chenxiao, Yue, P., Di, L., Wu, Z., 2018c. Automatic identification of center pivot irrigation systems from landsat images using convolutional neural networks. Agriculture 8, 147. https://doi.org/10.3390/agriculture8100147.

Zhang, X., Wang, Q., Chen, G., Dai, F., Zhu, K., Gong, Y., Xie, Y., 2018d. An object-based supervised classification framework for very-high-resolution remote sensing images

using convolutional neural networks. Remote Sens. Lett. 9, 373–382. https://doi.org/10.1080/2150704X.2017.1422873.
Zhao, W., 2017. Research on the deep learning of the small sample data based on transfer learning. AIP Conf. Proc. 020018. https://doi.org/10.1063/1.4992835.
Zhao, W., Du, S., 2016. Learning multiscale and deep representations for classifying remotely sensed imagery. ISPRS J. Photogramm. Remote Sens. 113, 155–165. https://doi.org/10.1016/j.isprsjprs.2016.01.004.
Zhao, W., Du, S., Wang, Q., Emery, W.J., 2017. Contextually guided very-high-resolution imagery classification with semantic segments. ISPRS J. Photogramm. Remote Sens. 132, 48–60. https://doi.org/10.1016/j.isprsjprs.2017.08.011.
Zhao, W., Zhou, G., Jun, Y., Zhang, X., Luo, L., 2015. On combining multiscale deep learning features for the classification of hyperspectral remote sensing imagery. Int. J. Remote Sens. 36, 3368–3379. https://doi.org/10.1080/2150704X.2015.1062157.
Zhu, F., Ma, Z., Li, X., Chen, G., Chien, J.T., Xue, J.H., Guo, J., 2019. Image-text dual neural network with decision strategy for small-sample image classification. Neurocomputing 328, 182–188. https://doi.org/10.1016/j.neucom.2018.02.099.